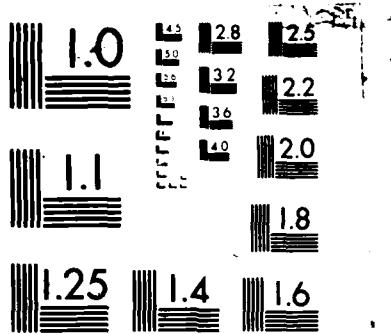


AD-A179 157 SEQUENTIAL CONVEXIFICATION IN REVERSE CONVEX AND 1/1
DISJUNCTIVE PROGRAMMING(U) CARNEGIE-MELLON UNIV
PITTSBURGH PA MANAGEMENT SCIENCES RESEAR.
UNCLASSIFIED E BALAS ET AL DEC 86 MSRR-531 F/G 12/2 NL





MICROCOPY RESOLUTION TEST CHART
© 1970 RAYMUND W. NEARNS, INC. A

AD-A179 157

DTIC EFILE COPY

(12)

SEQUENTIAL CONVEXIFICATION

IN REVERSE CONVEX AND DISJUNCTIVE PROGRAMMING

by

Egon Balas
Joseph M. Tama
and
Jørgen Tind*

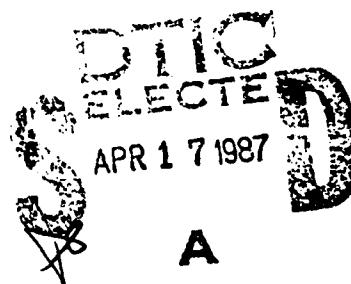
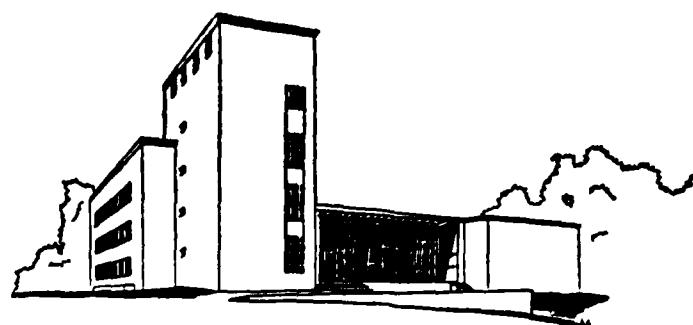
Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

This document has been approved
for public release and sale; its
distribution is unlimited.

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



87 4 14 051

(12)

SEQUENTIAL CONVEXIFICATION
IN REVERSE CONVEX AND DISJUNCTIVE PROGRAMMING

by

Egon Balas
Joseph M. Tama
and
Jørgen Tind*

December 1986

This document has been reviewed
for public release and it is
determined that its release is permitted.

The research underlying this report was supported by Grant ECS-8205425 of the National Science Foundation and Contract N00014-82-K-0329 NR047-607 with the Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

Management Science Research Group
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

*On leave from University of Aarhus, Denmark

Abstract

This paper is about a property of certain combinatorial structures, called sequential convexifiability, shown by Balas [1974, 1979] to hold for facial disjunctive programs. Sequential convexifiability means that the convex hull of a nonconvex set defined by a collection of constraints can be generated by imposing the constraints one by one, sequentially, and generating each time the convex hull of the resulting set. Here we extend the class of problems considered to disjunctive programs with infinitely many terms, also known as reverse convex programs, and give necessary and sufficient conditions for the solution sets of such problems to be sequentially convexifiable. We point out important classes of problems in addition to facial disjunctive programs (for instance, reverse convex programs with equations only) for which the conditions are always satisfied. Finally, we give examples of disjunctive programs for which the conditions are violated, and so the procedure breaks down.



0. Introduction

A procedure has been proposed in Balas [1974, 1979] for the sequential generation of the convex hull of feasible solutions to the system of constraints of a disjunctive programming problem. The idea behind the procedure in general terms is the repetition of a partial convex hull operation of the following kind.

Consider an optimization problem whose solution set is nonconvex. Its constraints may include linear or nonlinear inequalities, integrality constraints, logical conditions (implications, disjunctions, etc.) or whatever. Take a subsystem of these constraints and form the convex hull of its solution set. It is a *partial* convex hull relative to the convex hull of the set of solutions to the full system of constraints. Next, intersect this partial convex hull with the solution set of a second subsystem consisting of constraints not included in the first subsystem. Finally, form the convex hull of the intersection.

Under what conditions is the product of this partial convex hull operation the convex hull of solutions to the augmented subsystem formed by appending the constraints of the second subsystem to those of the first? Once aware of them, we can try to maintain these conditions through a number of repetitions of the operation, each enlarging the augmented subsystem by appending still unincorporated constraints until the complete system is formed. If we succeed, the result will be the convex hull of solutions to the complete system.

A procedure that generates the convex hull of a nonconvex set in this sequential manner will be called a *sequential convexification* procedure. A set whose convex hull can be generated in this manner will be called *sequentially convexifiable*.

Sequential convexifiability has both practical and theoretical implications. Sequential convexification is often a more efficient procedure than the

alternative of imposing all the constraints simultaneously and generating the convex hull of the resulting set. Besides, the procedure involves a number of iterations bounded by the number of constraints.

The question as to when precisely is a nonconvex set sequentially convexifiable was given a partial answer in the context of disjunctive programming by Balas [1974, 1979], in the form of a sufficient condition for the convex hull of a disjunctive set to be obtainable in the sequential manner outlined in the preceding paragraph. It requires the disjunctive program to be *facial*, a property (to be specified below) general enough to be shared by all pure or mixed 0-1 programming problems, but not shared by general integer programs. On the other hand, the question of a necessary condition for the validity of the sequential convexification procedure has remained open until this writing.

The present paper addresses this open question. However, in so doing it also places the question of sequential convexification in a somewhat more general context, in that, instead of disjunctive programs with finitely many terms in each disjunction, it considers the class of reverse convex programming problems, which can be viewed as disjunctive programs involving disjunctions with infinitely many terms. To be specific, we consider problems whose constraint set is of the form

$$g_i(x) \geq 0, \quad i = 1, 2, \dots, m$$

where, for each i , g_i is a convex function from R^n to R . In a disjunctive programming problem with finitely many terms in each disjunction, each g_i is either linear or piecewise linear and convex. In the former case it is called a *conjunctive constraint*. In the latter case it is called a *disjunctive constraint* and is written as a disjunction, each term of which is a linear constraint. This form of the constraint set is known as the *conjunctive*

normal form. Thus the class of reverse convex programs considered here corresponds to disjunctive programs stated in conjunctive normal form, whose disjunctions may have infinitely many terms.

The sequential convexification procedure, as proposed in Balas [1974, 1979], applies to *facial* disjunctive programs, i.e. those in which every term of every disjunction is a linear inequality that induces a face of the polyhedron defined by the conjunctive inequalities. It begins with the subsystem consisting of all the conjunctive constraints of the problem. It continues by appending one piecewise linear (or *disjunctive*) constraint to the growing subsystem with each repetition of the partial convex hull operation. The procedure to be considered here for the more general case of reverse convex programs retains this feature of introducing the constraints not included in the initial subsystem one by one. The main result of our paper is a necessary and sufficient condition for the partial convex hull operation to work as desired when only one reverse convex constraint is introduced at a time. The *Constraint Boundary Condition*, as we call it, is weaker than the condition of faciality when applied to a disjunctive program with finitely many terms. However, while faciality can be determined prior to the initiation of the procedure, this may not be true of our condition: it may be necessary to check each time the convex hull operation is repeated whether the *Constraint Boundary Condition* holds. On the other hand, since it is a necessary as well as sufficient condition for the partial convex hull operation to yield the desired product, the *Constraint Boundary Condition* imposes the weakest possible restriction on the constraints involved. The *Constraint Boundary Condition* involves pairs of points satisfying the first subsystem of constraints considered in a partial convex hull operation, during which the (partial) convex hull of the system is formed. Often this convex hull is a polytope (see

next paragraph), and in that case it is only necessary to consider pairs of vertices of the polytope, which makes the Constraint Boundary Condition finitely checkable.

The study of reverse convex programs goes back to Rosen (1966), who investigated such problems in a control theoretic setting. For a general discussion of important aspects of reverse convex programming, also called complementary convex programming, see Avriel (1976). Particularly relevant to the present paper is the study of Hillestad and Jacobsen (1980). They have shown that the convex hull of feasible solutions to a reverse convex programming problem is a polytope subject to certain differentiability and compactness assumptions. In this case, given a linear objective function, knowledge of the convex hull enables one to solve the problem by linear programming.

Our paper is organized as follows. In section 1 we state the Constraint Boundary Condition and show that it is necessary and sufficient for the success of the partial convex hull operation. We also state an alternative sufficient condition for the finite case, which is usually easier to check. In section 2 we give a formal statement of the sequential convexification procedure and show how its success depends upon the Constraint Boundary Condition being satisfied for every pair of sets in a certain sequence generated by the procedure. In section 3 we discuss some important special cases in which the Constraint Boundary Condition is always satisfied and hence the sequential convexification procedure always works. Finally, in section 4 we illustrate on a job shop scheduling problem the circumstances under which the sequential convexification procedure may break down, and the way these circumstances may be avoided by a slightly different problem formulation.

1. A Basic Relation

When will the partial convex hull operation yield the desired product? The partial convex hull operation is binary inasmuch as it operates upon two sets. On the one hand, there is the set of elements satisfying a subsystem of the constraints of an optimization problem. On the other hand, there is the set of elements satisfying an additional constraint. The operation consists in intersecting the convex hull of the former with the latter and forming the convex hull of the intersection. The desired product is the convex hull of elements satisfying the augmented system consisting of the first subsystem with the additional constraint appended to it. In this section that opening question is appropriately qualified, recast in formal terms, and answered.

The partial convex hull operation was conceived in connection with disjunctive programming problems. In that context the additional constraint is disjunctive. It has the form $g(x) \geq 0$ where $g(x) = \max_i(a_i x - b_i)$ and the linear functions $a_i x - b_i$ are finite in number. The complement of the solution set, $C = \{x \mid g(x) < 0\}$, if nonempty, is an open convex polyhedron of full dimension. Thus the imposition of the additional constraint can be viewed as the exclusion from the solution set of the elements in C . The set resulting from the intersection of multiple disjunctive constraints of this kind can be thought of as formed by discarding a union of open convex sets.

In the present context, the class of constraints eligible to fill the role of single additional constraint in the partial convex hull operation consists of reverse convex constraints. Each of these must be such that the set of elements not satisfying it is open, convex, and of full dimension. If C denotes this set, then it can be defined via a convex function

$g: \mathbb{R}^n \rightarrow \mathbb{R}$ as $C = \{x \mid g(x) < 0\}$. The solution set for the constraint is represented as its complement, $\sim C = \{x \mid g(x) \geq 0\}$.

The question of interest is now recast in formal terms as follows. Let P represent the set of solutions to the subsystem of constraints. Let $\sim C$ represent the set of elements satisfying the additional reverse convex constraint. The partial convex hull operation intersects $\text{conv } P$ and $\sim C$, which yields $(\text{conv } P)\sim C$, and forms the convex hull of the result. So the product of the partial convex hull operation is $\text{conv}[(\text{conv } P)\sim C]$. The set of solutions to the augmented system, on the other hand, is just the intersection of P and $\sim C$. So, in formal terms, for arbitrary P and for C open, convex and of full dimension the question of interest is to know when the following **Basic Relation** is true:

$$(1.1) \quad \text{conv}[(\text{conv } P)\sim C] = \text{conv}(P\sim C).$$

Because $(\text{conv } P)\sim C \supseteq P\sim C$, it is always the case that $\text{conv}[(\text{conv } P)\sim C] \supseteq \text{conv}(P\sim C)$. The main result of our paper is that the reverse inclusion holds if and only if the following condition, called the **Constraint Boundary Condition**, is satisfied.

$$(1.2) \quad \text{If } x \in P \cap C \text{ and } y \in P\sim C, \text{ then } [x,y] \cap \text{bd } C \in \text{conv}(P\sim C).$$

Here $[x,y]$ denotes the line segment between x and y and $\text{bd } C$ denotes the boundary of C in the affine space spanned by C . Observe that since C is full dimensional, $[x,y] \cap \text{bd } C \neq \emptyset$.

Theorem 1.1. *The sets P and C satisfy the Basic Relation (1.1) if and only if they satisfy the Constraint Boundary Condition (1.2).*

The "if" part, which is the main content of Theorem 1.1, says that in order to ascertain that all points of $(\text{conv } P)\sim C$ belong to $\text{conv}(P\sim C)$, it is

sufficient to check that all those points on the boundary of C that are the convex combination of just two points of P , belong to $\text{conv}(P^\sim C)$. The "only if" part states the easily verifiable converse.

The proof of Theorem 1.1 will make use of the following auxiliary result.

Lemma 1.2. *Let $x_i, y_j \in \mathbb{R}^n$, $i = 1, \dots, h$; $j = 1, \dots, k$; and let $z_{ij} \in [x_i, y_j]$ be given for all pairs i, j . Then for all $x \in \text{conv}\{x_1, \dots, x_h\}$ and $y \in \text{conv}\{y_1, \dots, y_k\}$, the set $[x, y] \cap \text{conv}\{z_{ij} \mid i = 1, \dots, h; j = 1, \dots, k\}$ is nonempty.*

Proof. By contradiction. If $[x, y] \cap \text{conv}\{z_{ij} \mid i=1, \dots, h; j=1, \dots, k\} = \emptyset$, then there exists a hyperplane which strongly separates the line segment $[x, y]$ from the polytope $\text{conv}\{z_{ij} \mid i=1, \dots, h; j=1, \dots, k\}$ (Cf. Rockafellar (1970), Theorem 11.4). Hence the hyperplane defines an open halfspace H^+ which contains $[x, y]$ but does not contain any z_{ij} . On the other hand, because $x \in H^+$ and $y \in H^+$ we must have at least one x_i and at least one y_j in H^+ . But for this x_i and this y_j we must have $[x_i, y_j]$, and hence z_{ij} , in H^+ , which is a contradiction. \square

Proof of Theorem 1.1. Assume that (1.1) holds. Let $x \in P \cap C$ and $y \in P^\sim C$, and let $w = [x, y] \cap \text{bd } C$. Because $x \in P$ and $y \in P$, $w \in \text{conv } P$. Moreover, as C is open, $C \cap \text{bd } C = \emptyset$ and $w \notin C$. Thus $w \in (\text{conv } P)^\sim C \subseteq \text{conv}[(\text{conv } P)^\sim C] = \text{conv}(P^\sim C)$, where the equality holds by assumption.

Conversely, assume that (1.2) holds. As remarked above, it is sufficient to show that $\text{conv}[(\text{conv } P)^\sim C] \subseteq \text{conv}(P^\sim C)$. So, suppose that z is a point of $\text{conv}[(\text{conv } P)^\sim C]$. (If $\text{conv}[(\text{conv } P)^\sim C] = \emptyset$, then so, too, for the included set, $\text{conv}(P^\sim C) = \emptyset$, and we need not continue.) Then, by Caratheodory's Theorem, z can be expressed as a convex combination of finitely many points $r \in (\text{conv } P)^\sim C$. In turn each such r can be expressed as a convex combination of finitely many points of P . For each such r , because $r \notin C$ and C is convex,

the points of P in the convex combination for r must include at least one point in $P \sim C$. On the other hand, each r which is a convex combination solely of points in $P \sim C$ must belong to $\text{conv}(P \sim C)$. In order to conclude that $z \in \text{conv}(P \sim C)$, we need only show that those r which are not convex combinations of points solely in $P \sim C$ are also in $\text{conv}(P \sim C)$. So assume that r can be expressed as a convex combination of the points $x_1, \dots, x_h, y_1, \dots, y_k$ where $h, k \geq 1$ and $x_i \in P \cap C$, each $y_j \in P \sim C$. For each pair x_i, y_j define $z_{ij} = [x_i, y_j] \cap \text{bd } C$. Hence $z_{ij} \in \text{cl } C$ (cl denotes the closure). By the Constraint Boundary Condition (1.2), $z_{ij} \in \text{conv}(P \sim C)$. Furthermore, by collection of terms, r can in this instance be expressed as a convex combination of just two points $x \in \text{conv}\{x_1, \dots, x_h\}$ and $y \in \text{conv}\{y_1, \dots, y_k\}$. That is, r is on the closed line segment $[x, y]$. By Lemma 1.3 there exists $w \in [x, y] \cap \text{conv}\{z_{ij} \mid i=1, \dots, h; j=1, \dots, k\}$. So $[x, y] = [x, w] \cup [w, y]$, and either $r \in [x, w]$ or $r \in [w, y]$. Now $w \in \text{cl } C$ as $z_{ij} \in \text{cl } C$ for all i, j . Similarly, $w \in \text{conv}(P \sim C)$ as $z_{ij} \in \text{conv}(P \sim C)$ for all i, j . Since $x \in C = \text{int } C$ (int denotes interior) and $w \in \text{cl } C$, we have $[x, w] \subseteq C$, while, as w and y are in $\text{conv}(P \sim C)$, so too is $[w, y]$. The fact that $r \notin C$ then implies $r \in [w, y]$ and $r \in \text{conv}(P \sim C)$. So each r is in $\text{conv}(P \sim C)$, and z , as a convex combination of them, is there as well. Therefore, $\text{conv}[(\text{conv } P) \sim C] \subseteq \text{conv}(P \sim C)$. \square

For the case of a disjunctive program with finitely many terms in each disjunction, we give a sufficient condition for the Basic Relation to hold, which is often easier to check than (1.2). Let $H_i^- = \{x \mid d^T x \leq d_{i0}\}$, $i \in Q$, be finitely many halfspaces of the space containing P .

Theorem 1.3. The Basic Relation

$$(1.1) \quad \text{conv}((\text{conv}P)\sim C) = \text{conv}(P\sim C),$$

where $\sim C = \bigcup_{i \in Q} H_i^-$, is satisfied if the relation

$$(1.3) \quad (\text{conv}P) \cap H_i^- = \text{conv}(P \cap H_i^-)$$

is satisfied for every $i \in Q$.

Proof. Suppose (1.3) holds for every $i \in Q$. Then

$$\begin{aligned} \text{conv}((\text{conv}P)\sim C) &= \text{conv}((\text{conv}P) \cap (\bigcup_{i \in Q} H_i^-)) \\ &= \text{conv} \bigcup_{i \in Q} ((\text{conv}P) \cap H_i^-) \\ &= \text{conv} \bigcup_{i \in Q} \text{conv}(P \cap H_i^-) \\ &= \text{conv} \bigcup_{i \in Q} (P \cap H_i^-) \\ &= \text{conv}(P \cap (\bigcup_{i \in Q} H_i^-)) \\ &= \text{conv}(P \sim C). \end{aligned}$$

Here the first and last equations were obtained by using $\sim C = \bigcup_{i \in Q} H_i^-$, the second and next to last equations use the fact that \cap is distributive with respect to \cup , the third equation follows from (1.3) (which is assumed to hold), while the fourth equation uses the obvious relation $\text{conv}(\text{conv}S \cup \text{conv}T) = \text{conv}(S \cup T)$, true for arbitrary sets S, T . \square

While (1.3) is sufficient for (1.1) to hold, it is not necessary. To see this, let $P \subseteq \mathbb{R}^2$, $P = \{(1,1), (-1,1)\}$ and $\sim C = H_1^- \cup H_2^-$, where $H_1^- = \{x \in \mathbb{R}^2 \mid x_1 + x_2 \geq 3/2\}$, $H_2^- = \{x \in \mathbb{R}^2 \mid -x_1 + x_2 \geq 3/2\}$. We have $P \sim C = P$, $\text{conv}(P \sim C) = \text{conv } P = \text{conv}((\text{conv}P)\sim C)$, i.e. (1.1) holds. On the other hand, $P \cap H_1^- = \{(1,1)\} = \text{conv}(P \cap H_1^-)$, whereas $\text{conv } P = \{x \in \mathbb{R}^2 \mid x = (1,1)\lambda + (-1,1)(1-\lambda), 0 \leq \lambda \leq 1\}$, and

$$\begin{aligned} (\text{conv}P) \cap H_1^- &= \left\{ x \in \mathbb{R}^2 \mid x = (1,1)\lambda + (-1,1)(1-\lambda), 0 \leq \lambda \leq 1 \right. \\ &\quad \left. x_1 + x_2 \geq 3/2 \right\} \\ &= \{x \in \mathbb{R}^2 \mid x = (1,1)\lambda + (-1,1)(1-\lambda), 3/4 \leq \lambda \leq 1\}. \end{aligned}$$

Clearly, $\text{conv}(P \cap H_1^-) \neq (\text{conv}P) \cap H_1^-$, i.e. (1.3) does not hold.

Relation (1.3) is equivalent to a condition that has an appealing geometric interpretation. We will say that H_i^- (or H_i , where $H_i = \text{bd } H_i^-$) satisfies the **Window Condition** with respect to P if the following is true:

(1.4) If $x \in P \cap H_i^-$ and $y \in P \cap H_i^-$, then $[x,y] \cap H_i^- \subseteq \text{conv}(P \cap H_i^-)$.

Requirement (1.4) can be interpreted as having the set $\text{conv}(P \cap H_i^-)$ act as a "window" through which every pair of points lying in P but on opposite sides of H_i^- , can "see each other."

Theorem 1.4. *Equation (1.3) holds if and only if the Window Condition (1.4) holds.*

Proof. Applying Theorem 1.1 to P and C in the case when $C = \text{int } H_i^+$ ($= \sim H_i^-$), we find that (1.3) is in this case the same as (1.1), while the Constraint Boundary Condition (1.2) becomes

(1.2') If $x \in P \cap H_i^-$ and $y \in P \cap H_i^-$, then $([x,y] \cap H_i^-) \subseteq \text{conv}(P \cap H_i^-)$.

The Window Condition (1.4) differs from (1.2') only in that it replaces H_i^- with H_i on the righthand side of the last expression. Now if (1.4) holds then (1.2') clearly does, so all that remains to be shown is that (1.2') implies (1.4). Let $w := [x,y] \cap H_i$. Since $w \in H_i \cap \text{conv}(P \cap H_i^-)$ (from (1.2')), w lies on the face of $\text{conv}(P \cap H_i^-)$ contained in the hyperplane H_i ; hence $w \in \text{conv}(P \cap H_i^-)$. ||

Corollary 1.5. *Let P be any set, and $\sim C = \bigcup_{i \in Q} H_i^-$. Then the Basic Relation (1.1) holds if the Window Condition (1.4) holds for every hyperplane H_i , $i \in Q$.*

Proof. Follows from Theorems 1.3 and 1.4. ||

2. The Procedure For Sequential Generation of the Convex Hull

In this section we give a formal description of the sequential convexification procedure. This confirms that the procedure succeeds if upon each repetition of the partial convex hull operation the two sets subjected to the operation satisfy the Constraint Boundary Condition. In addition, we provide a sufficient condition for the success of the procedure in the more general version which adds constraints sequentially, but not one at a time.

The procedure is designed for problems which include some reverse convex constraints, each of which is such that the set of points satisfying the constraint has a complement that is open, convex and of full dimension. If the problem has other kinds of constraints as well, these other constraints can be included in the initial subsystem.

The procedure consists in a number of iterations each of which uses the partial convex hull operation to activate a single one of the constraints so far left out. The constraint is made active by intersecting the set resulting from the preceding iteration with the set of points satisfying the constraint, and forming the convex hull of the intersection. The procedure continues until all constraints have been made active. The desired result of the final iteration is the convex hull of the set of solutions to all the constraints of the problem.

Our approach is best understood by viewing the activation of a constraint at an iteration of the procedure as a narrowing of the feasible region by excluding from consideration an open, convex, full-dimensional set. The set excluded is the complement of the set of points satisfying the constraint. The procedure could be viewed as consisting of the sequential exclusion of such sets. From this perspective, the procedure can be described as follows.

Given an initial subsystem of the constraints, denote the set of solutions to this subsystem F_1 . We assume $F_1 \in \mathbb{R}^n$. Given that there are m remaining constraints, each satisfied by a set of points whose complement is open, convex, and of full dimension, order the constraints and denote the corresponding open, convex complements C_1, \dots, C_m . Set $C = \bigcup_{j=1}^m C_j$. The object of the procedure is the formation of the convex hull of the set F given by

$$F = F_1 \sim C = F_1 \sim \bigcup_{j=1}^m C_j.$$

Alternatively, F can be described by successive exclusion of the sets C_j :

$$F = ((\dots((F_1 \sim C_1) \sim C_2) \dots) \sim C_m).$$

At the j^{th} iteration the procedure forms the set $F_{j+1} \in \mathbb{R}^n$ defined recursively as $F_{j+1} = (\text{conv } F_j) \sim C_j$, $j = 1, \dots, m$, and then takes its convex hull. $\text{conv } F_{j+1}$ will be compared below to the convex hull of the set G_{j+1} defined recursively as

$$G_1 = F_1$$

$$G_{j+1} = G_j \sim C_j, \quad j = 1, \dots, m.$$

Observe that $G_{m+1} = F$.

Theorem 2.1. Let F_1 be an arbitrary set, and let C_j , $j = 1, \dots, m$, be open convex sets of full dimension, ordered into an arbitrary but fixed sequence. Then $\text{conv } F_j = \text{conv } G_j$ for $j = 2, \dots, m+1$ if and only if the Constraint Boundary Condition holds for each pair G_j, C_j , $j = 1, \dots, m$. If this is the case, then in particular $\text{conv } F_{m+1} = \text{conv } F$.

Proof. Assume that the Constraint Boundary Condition (CBC) holds for all pairs G_j, C_j , $j = 1, \dots, m$. Apply the procedure to F_1, F_2, \dots, F_m . After the first step we have:

$$\begin{aligned}
\text{conv } F_2 &= \text{conv}[(\text{conv } F_1) \sim C_1] \text{ (by definition of } F_2) \\
&= \text{conv}[(\text{conv } G_1) \sim C_1] \text{ (since } F_1 = G_1) \\
&= \text{conv}(G_1 \sim C_1) \quad \text{(since } G_1, C_1 \text{ satisfy the CBC)} \\
&= \text{conv } G_2 \quad \text{(by definition of } G_2).
\end{aligned}$$

The proof continues by induction. By hypothesis, after step $j - 1$ we have

$\text{conv } F_j = \text{conv } G_j$. Therefore, after step j we have:

$$\begin{aligned}
\text{conv } F_{j+1} &= \text{conv}[(\text{conv } F_j) \sim C_j] \\
&= \text{conv}[(\text{conv } G_j) \sim C_j] \text{ (by the induction hypothesis)} \\
&= \text{conv}(G_j \sim C_j) \quad \text{(since } G_j, C_j \text{ satisfy the CBC)} \\
&= \text{conv } G_{j+1}.
\end{aligned}$$

After m steps, since $G_{m+1} = F$, we have: $\text{conv } F_{m+1} = \text{conv } F$.

Conversely, if the Constraint Boundary Condition fails for a pair G_j, C_j , then:

$$\begin{aligned}
\text{conv } F_{j+1} &= \text{conv}[(\text{conv } F_j) \sim C_j] \\
&\supseteq \text{conv}[(\text{conv } G_j) \sim C_j] \text{ (since } F_j \supseteq G_j) \\
&\not\supseteq \text{conv}(G_j \sim C_j) \quad \text{(by Theorem 1.4)} \\
&= \text{conv } G_{j+1}. \square
\end{aligned}$$

Two comments are in order at this point, each of which reveals a certain weakness of our procedure; weaknesses that are overcome in the important special cases to be discussed in the next section.

First, note that for the sequential convexification procedure to be valid, Theorem 2.1 requires (1.2) to hold for each pair G_j, C_j , where the set G_j is in the role of P and C_j in that of C . The feature we wish to point out here is that although $\text{conv } G_j = \text{conv } F_j$, where F_j is defined recursively as $\text{conv } (F_{j-1} \sim C_{j-1})$, it is *not sufficient* for condition (1.2) to hold for each pair F_j, C_j . In other words, while the sequential convexification procedure is a recursive application of the partial convex hull operation to each pair

F_j , C_j , the validity of the procedure hinges on a condition that at each step of the procedure reaches back beyond F_j , to the set G_j . This of course makes the condition harder to check.

Second, note that the condition given in Theorem 2.1 for the validity of the sequential convexification procedure is tied to a certain ordering of the constraints, hence of the pairs G_j , C_j . The procedure may be valid if the constraints are imposed in a particular order and invalid if a different ordering is used. This makes a negative outcome often inconclusive: in order to establish that the procedure is inapplicable to a certain problem, one might have to explore exponentially many sequences.

Fortunately, both of these shortcomings disappear in certain special cases which are rather important; they will be discussed in the next section.

Now we turn to the case of a disjunctive program with finitely many terms in each disjunction, and give a sufficient condition for the validity of the sequential convexification procedure.

Corollary 2.2. Let F_1 and C_j be as in Theorem 2.1, with $C_j = \bigcup_{i \in Q_j} H_{ji}$, $j = 1, \dots, m$, where each Q_j is finite and each H_{ji} is a closed halfspace. Then $\text{conv } F_j = \text{conv } G_j$ for $j = 2, \dots, m+1$ if for $j = 1, \dots, m$ each hyperplane H_{ji} ($= \text{bd } H_{ji}$) satisfies the Window Condition (1.4) with respect to G_j .

Proof. Analogous to the proof of sufficiency in Theorem 2.1, with condition (1.2) replaced by (1.4). \square

In the Introduction we also outlined a more general version of the procedure which involves the possibility of activating several constraints at each iteration. An iteration of this version consists of intersecting the result of the preceding iteration with the set of solutions satisfying several of the as yet unactivated constraints. Then the convex hull of the intersection is

formed. Next we show that if there is an ordering of the m constraints not included in the initial subsystem such that each pair G_j, C_j satisfies the Constraint Boundary Condition, then a procedure which activates any number of the still inactive constraints at each iteration will be successful. However, if several constraints are activated at once, then they must be selected consecutively according to their original ordering.

Theorem 2.3. *Let F_1 be an arbitrary set, and let $C_j, j = 1, \dots, m$, be open convex sets of full dimension. If the Constraint Boundary Condition holds for each pair $G_j, C_j, j = 1, \dots, k \leq m$, then*

$$\text{conv}[(\text{conv } F_1)^\sim (\bigcup_{j=1}^k C_j)] = \text{conv } G_{k+1}.$$

Proof. On the one hand, $(\text{conv } F_1)^\sim (\bigcup_{j=1}^k C_j) \supseteq F_1^\sim (\bigcup_{j=1}^k C_j)$. Therefore

$$\text{conv}[(\text{conv } F_1)^\sim (\bigcup_{j=1}^k C_j)] \supseteq \text{conv}[F_1^\sim (\bigcup_{j=1}^k C_j)] = \text{conv } G_{k+1}.$$

On the other hand, the opposite inclusion follows by induction. By Proposition 2.1 it holds for $k = 1$. Suppose it holds for $k = i - 1$, $2 \leq i \leq m$, and set $k = i$. Then

$$\begin{aligned} \text{conv}[(\text{conv } F_1)^\sim (\bigcup_{j=1}^k C_j)] &= \text{conv}[(\text{conv } F_1)^\sim (\bigcup_{j=1}^{k-1} C_j)^\sim C_k] \\ &\subseteq \text{conv}[\text{conv}[(\text{conv } F_1)^\sim (\bigcup_{j=1}^{k-1} C_j)]^\sim C_k] \\ &= \text{conv}[(\text{conv } G_{k-1})^\sim C_k] \quad (\text{by hypothesis}) \\ &= \text{conv}(G_{k-1}^\sim C_k) \quad (\text{by the CBC}) \\ &= \text{conv } G_{k+1}. \quad \square \end{aligned}$$

3. Some Special Cases

In this section we discuss some of the results of the previous sections as they apply to certain special cases.

First, there are situations in which the Constraint Boundary Condition (1.2) holds trivially. One case of this kind occurs when $P \cap C = \emptyset$. Another one when $P \cap C = P$. In the first case there is no $x \in P \cap C$, in the second one there is no $y \in P \cap C$; so in both cases the condition (1.2) holds trivially and thus from Theorem 1.1, the Basic Relation (1.1) also holds.

A very important case, with interesting subcases, is the one when $P \not\subseteq C$ but $P \subseteq \text{cl } C$. Consider for instance a nonlinear programming problem whose only nonlinear constraints are equations involving convex functions, i.e. are of the form $g_i(x) = 0$, $i = 1, \dots, m$, with each g_i a convex function such that $\{x \mid g_i(x) < 0\} \neq \emptyset$. We can then define F_1 (i.e. the initial set P) as the set of points satisfying the linear constraints plus the nonlinear inequalities $g_i(x) \leq 0$, $i = 1, \dots, m$; and for $i = 1, \dots, m$ let $C_i = \{x \mid g_i(x) < 0\}$, i.e. $\sim C_i = \{x \mid g_i(x) \geq 0\}$. On the one hand, this makes F_1 convex, hence the procedure is relatively easy to start. On the other hand, with the chosen definition of F_1 and of the sets C_i , at any stage of the procedure if the current set G_j is P and the next constraint C_j to be activated is C , then $y \in P \cap C$ implies $y \in \text{bd } C$ and so for any $x \in P \cap C$, we have $[x,y] \cap \text{bd } C = \{y\}$. The Constraint Boundary Condition (1.2) is thus necessarily satisfied. In other words, for optimization problems whose only nonlinear constraints are equations involving convex functions, the sequential convexification procedure always works; and it does so irrespective of the sequence in which the constraints are activated.

Another instance when $P \not\subseteq C$ but $P \subseteq \text{cl } C$ occurs in the case of facial disjunctive programs, i.e. problems for which F_1 (the initial set P) is a

polyhedron given by a set of linear inequalities, and each disjunctive constraint is of the form $\sim C_j = \bigcup_{i \in Q_j} H_{ji}$, with Q_j finite and H_{ji} a closed halfspace such that $H_{ji} \cap F_1$ is a face of F_1 . This situation is again easy to recognize, since all one has to do is to check for each halfspace H_{ji} that the inequality defining the opposite closed halfspace, H_{ji}^+ , is among the inequalities defining F_1 . As in the situation discussed earlier, the sequential convexification procedure is then applicable without any need for checking the Constraint Boundary Condition and the constraints may be imposed in any order whatsoever.

As already mentioned, facial disjunctive programs include all pure or mixed integer 0-1 programs. The fact that their convex hulls can be generated sequentially in n steps, where n is the number of disjunctions, makes it possible to develop cutting plane procedures whose finiteness proof uses this property (see Jeroslow [1980] for one such procedure).

For general reverse convex programs, $\text{conv } G_j$ is often a polytope (this is subject, as already mentioned, to certain differentiability and compactness assumptions; see Hillestad and Jacobsen [1980] for details). In this case $\text{conv } G_j$ is generated by a finite number of so-called quasivertices, and the checking of the Constraint Boundary Condition can be restricted to pairs x, y of such quasivertices.

4. A Job Shop Scheduling Example.

Consider the problem of scheduling the processing on n jobs on one machine. If L_j denotes the earliest time job j can be started, d_{ij} the minimum time lapse between the starting of job i and that of job j (in that order) on the machine, and (the variable) t_j denotes the actual starting time

of job j , then for whatever objective function, the constraints of the problem are

$$(4.1) \quad t_j \geq L_j, \quad j \in N$$

$$t_j - t_i \geq d_{ij} \vee t_i - t_j \geq d_{ji}, \quad i, j \in N, \quad i \neq j$$

where $N = \{1, \dots, n\}$. This is a disjunctive set, and it is easy to see that it is not facial. Indeed, faciality would require that the conditions

$$\begin{aligned} t_j - t_i &\leq d_{ij} \\ -t_j + t_i &\leq d_{ji} \quad i, j \in N, \quad i \neq j \end{aligned}$$

be added to the constraints of (4.1), which would obviously change the problem, actually make it infeasible in most cases. So whether the sequential convexification procedure works for this class of problems is an open question. We will show by way of a small example that it does not work in general.

Example. Let $n = 3$, $L_i = 0$, $j = 1, 2, 3$, and

$$(d_{ij}) = \begin{pmatrix} * & 2 & 4 \\ 3 & * & 6 \\ 2 & 2 & * \end{pmatrix}.$$

Then $F_1 = \mathbb{R}_+^3$ (the positive orthant of \mathbb{R}^3), and each disjunctive constraint set is of the form $\sim C_{ij}$, where

$$C_{ij} = \{t \in \mathbb{R}^3 \mid -d_{ji} \leq t_j - t_i \leq d_{ij}\}, \quad i, j = 1, 2, 3; \quad i \neq j\}.$$

If we impose the constraints in the order $\sim C_{12}$, $\sim C_{23}$, $\sim C_{13}$, the procedure generates the set

$$\text{conv } F_4 = \left\{ t \in \mathbb{R}_+^3 \mid \begin{array}{l} 2t_1 + 3t_2 \geq 6 \\ 3t_2 + t_3 \geq 6 \\ 2t_1 + t_3 \geq 4 \\ 4t_1 + 9t_2 + 2t_3 \geq 26 \\ t_1 + 18t_2 + 5t_3 \geq 38 \end{array} \right\}.$$

The sets F_2 and F_3 consist of those $t \in \mathbb{R}_+^3$ satisfying the first constraint, and the first two constraints, respectively, of F_4 . So the last step adds three constraints.

Now F_4 is not the convex hull of $F := F_1 \sim (\bigcup_{i,j=1}^3 C_{ij})$. The latter has, besides the three facets defined by the nonnegativity constraints and the three facets defined by inequalities with positive coefficients for two of the three variables, another four facets defined by inequalities with positive coefficients for all three variables (see Balas [1984] for an explanation of why this is so), and is in fact

$$\text{conv } F = \left\{ t \in \mathbb{R}_+^3 \mid \begin{array}{l} 2t_1 + 3t_2 \geq 6 \\ 3t_2 + t_3 \geq 6 \\ 2t_1 + t_3 \geq 4 \\ 9t_1 + 11t_2 + 5t_3 \geq 62 \\ 3t_1 + t_2 + t_3 \geq 10 \\ 14t_1 + 21t_2 + 10t_3 \geq 112 \\ 2t_1 + 33t_2 + 10t_3 \geq 76 \end{array} \right\}.$$

So the sequential convexification procedure does not work in this case; namely, it breaks down at the third iteration, when we take the convex hull of $(\text{conv } F_3) \sim C_{13}$. Indeed

$$G_3 = \left\{ t \in \mathbb{R}_+^3 \mid \begin{array}{l} t_2 - t_1 \geq 2 \vee t_1 - t_2 \geq 3 \\ t_3 - t_2 \geq 6 \vee t_2 - t_3 \geq 2 \end{array} \right\},$$

$$C_{13} = \left\{ t \in \mathbb{R}^3 \mid -2 < t_3 - t_1 < 4 \right\}.$$

Consider the points $x = (0, 2, 0)$ and $y = (5, 2, 0)$. Both are in G_3 , but $x \in C_{13}$, while $y \notin C_{13}$. Now $[x, y] \cap \text{bd } C_{13} = (2, 2, 0)$, and it is easy to check that $(2, 2, 0) \notin \text{conv}(G_3 \sim C_{13}) = \text{conv } F$, as $(2, 2, 0)$ violates each of the last four inequalities of $\text{conv } F$.

Our example illustrates the fact that for a nonfacial disjunctive program the sequential convexification procedure may not work. On the other hand, any bounded disjunctive program can be restated as a (pure or mixed integer) 0-1 program, by introducing one 0-1 variable for every term of every disjunction. By becoming a 0-1 program, the disjunctive program acquires the property of faciality, and lends itself -- in its new form -- to the sequential convexification procedure. This is a rather intriguing fact, in view of the circumstance that the inequalities containing the 0-1 variables in many instances have no constraining power in the linear programming relaxation of the problem. A case in point is the above discussed job shop scheduling problem, whose constraint set in the standard 0-1 programming formulation becomes

$$\begin{aligned} t_j &\geq L_j, & j \in N \\ t_j - t_i + My_{ij} &\geq d_{ij} \\ -t_j + t_i - My_{ij} &\geq d_{ji} - M \\ y_{ij} &\in \{0, 1\}, i, j \in N, i \neq j. \end{aligned}$$

Here M is a sufficiently large number to serve as an upper bound on any of the differences $t_j - t_i$. It is easy to see that if the 0-1 condition on each y_{ij} is relaxed to $0 \leq y_{ij} \leq 1$, the constraints involving the variables y_{ij} are ineffective in the sense that the continuous variables t_j are only constrained by their lower bounds: for any values $t_j \geq L_j$, $j \in N$, there exist associated values of y_{ij} satisfying $0 \leq y_{ij} \leq 1$ and all the remaining inequalities. Yet the disjunctive constraints are now of the form $y_{ij} \leq 0 \vee y_{ij} \geq 1$, with each y_{ij} constrained in addition by $0 \leq y_{ij} \leq 1$. Clearly, in this formulation our disjunctive program is facial and thus the sequential convexification procedure is valid for it.

It should be noted, however, that although from this particular point of view there is an advantage in representing the disjunctions by 0-1 variables, for other purposes a direct approach often yields results that cannot be obtained, or are much harder to obtain, by the use of 0-1 variables (see Balas [1985] for an illustration on the case of job shop scheduling).

References

M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice Hall, Englewood Cliffs, N.J. (1976).

E. Balas, "Disjunctive Programming: Properties of the Convex Hull of Feasible Points," MSRR No. 348, Carnegie Mellon University, 1974.

E. Balas, "Disjunctive Programming," *Annals of Discrete Mathematics*, 5, (1979), P. 3-51.

E. Balas, "On the Facial Structure of Scheduling Polyhedra," *Mathematical Programming Study* 24, 1985, p. 179-218.

R. J. Hillestad and S. E. Jacobsen, "Reverse Convex Programming," *Applied Mathematics and Optimization*, 6, (1980), p. 63-78.

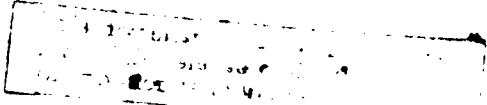
R. G. Jeroslow, "A Cutting Plane Game and Its Algorithms," D. P. 7724, CORE, Louvain, June 1977.

R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, N.J. (1970).

J. B. Rosen, "Iterative Solution of Nonlinear Optimal Control Problems," *SIAM J. of Control*, 4, (1966), p. 223-244.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER MSRR 531	2. GOVT ACCESSION NO. <i>7-111-17</i>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SEQUENTIAL CONVEXIFICATION IN REVERSE CONVEX AND DISJUNCTIVE PROGRAMMING		5. TYPE OF REPORT & PERIOD COVERED Technical Report, December 1986
6. AUTHOR(s) Egon Balas, Joseph M. Tama, Jørgen Tind		7. CONTRACT OR GRANT NUMBER(s) N00014 82 K 0329
8. PERFORMING ORGANIZATION NAME AND ADDRESS Graduate School of Industrial Administration Carnegie Mellon University Pittsburgh PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
9. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research (Code 434) Arlington VA 22217		12. REPORT DATE December 1986
11. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 23
14. SECURITY CLASS. (or info. report)		
15. DECLASSIFICATION/DEMANDING SCHEDULE		
16. DISTRIBUTION STATEMENT (of the Report)		
		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) reverse convex programming facial disjunctive programs sequential convexification		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper is about a property of certain combinatorial structures, called sequential convexifiability, shown by Balas (1974, 1979) to hold for facial disjunctive programs. Sequential convexifiability means that the convex hull of a nonconvex set defined by a collection of constraints can be generated by imposing the constraints one by one, sequentially, and generating each time the convex hull of the resulting set. Here we extend the class of problems considered to disjunctive programs with infinitely many terms, also known as _____ (cont.)		

reverse convex programs, and give necessary and sufficient conditions for the solution sets of such problems to be sequentially convexifiable. We point out important classes of problems in addition to facial disjunctive programs (for instance, reverse convex programs with equations only) for which the conditions are always satisfied. Finally, we give examples of disjunctive programs for which the conditions are violated, and so the procedure breaks down.

E

V

D

5 —

87

DTI C